CLAIMS

What is claimed is:

1. A method for optimizing database transaction performance in a database transaction processor having transaction services threads capable of being in active, non-active, and waiting states, said method comprising:

(a) adding a database change to a top of a queue; and

(b) starting a non-active transaction service thread conditioned upon less than a predetermined maximum number of transaction service threads being present.

2. A method in accordance with Claim 1 further comprising removing a non-active transaction service thread conditioned upon there being more than the lesser of said predetermined maximum number or a dynamically determined optimum number of transaction service threads present.

3. A method in accordance with Claim 1 further comprising changing a waiting transaction service thread to a non-active state, conditioned upon not less than a predetermined maximum number of transaction service threads being present.

4. A method in accordance with Claim 1 further comprising:

changing the state of a non-active transaction service thread to active conditioned upon there being a database change in the queue; and

using the active transaction service thread:

removing a bottom database change from the queue;

performing database changes specified by the removed database change; and

placing the transaction service thread into the non-active state.

5. A method in accordance with Claim 1 wherein said adding a non-active transaction service thread is further conditioned upon there being less than a dynamically determined optimum number of transaction service threads.

6. A method in accordance with Claim 6 further comprising determining said dynamically determined optimum number of transaction service threads dependent upon a ratio of an arrival rate of database changes to the queue divided by a service time of items removed from the queue.

7. A method in accordance with Claim 1 wherein adding a database change to a top of a queue further comprises adding a corresponding set of one or more interested listeners to said queue.

8. A method in accordance with Claim 7 further comprising:

changing the state of a non-active transaction service thread to active conditioned upon there being a database change in the queue; and

using the active transaction service thread:

removing a bottom database change and the corresponding set of interested listeners from the queue;

notifying said interested listeners that the removed database change has begun;

performing and committing database changes specified by the removed database change, conditioned upon obtaining locks necessary for transactions required for the removed database change;

notifying said interested listeners of a completion status of the removed database change; and

placing the transaction service thread into the non-active state.

9. A computing apparatus having a central processing unit operatively coupled to a memory including a database change queue, said apparatus configured to process a plurality of treads capable of being in active, non-active, and waiting states, said apparatus further configured to:

(a) add a database change to a top of the database change queue; and

(b) add a non-active transaction service thread, or change a waiting transaction service thread to a non-active state, conditioned upon whether there are less than, or not less than a predetermined maximum number of transaction service threads present, respectively.

10. An apparatus in accordance with Claim 9 further configured to remove a non-active transaction service thread conditioned upon there being more than the lesser of said predetermined maximum number or a dynamically determined optimum number of transaction service threads present, and to determine said dynamically determined optimum number of transaction service threads dependent upon a ratio of an arrival rate of

database changes to the queue divided by a service time of items removed from the queue.

11. An apparatus in accordance with Claim 9 further configured to:

change the state of a non-active transaction service thread to active conditioned upon there being a database change in the queue; and

using the active transaction service thread:

remove a bottom database change from the queue;

perform database changes specified by the removed database change; and

place the transaction service thread into the non-active state.

12. An apparatus in accordance with Claim 9 configured to further condition said adding a non-active transaction service thread upon there being less than a dynamically determined optimum number of transaction service threads, and to determine said dynamically determined optimum number of transaction service threads dependent upon a ratio of an arrival rate of database changes to the queue divided by a service time of items removed from the queue.

13. An apparatus in accordance with Claim 9 further configured to add a corresponding set of one or more interested listeners to the top of said queue along with said database change.

14. An apparatus in accordance with Claim 13 further configured to:

change the state of a non-active transaction service thread to active conditioned upon there being a database change in the queue; and

using the active transaction service thread:

remove a bottom database change and the corresponding set of interested listeners from the queue;

notify said interested listeners that the removed database change has begun;

perform and committing database changes specified by the removed database change, conditioned upon obtaining locks necessary for transactions required for the removed database change;

notify said interested listeners of a completion status of the removed database change; and

place the transaction service thread into the non-active state.

15. A machine-readable medium or media having recorded thereon instructions configured to instruct a computing apparatus having a central processing unit operatively coupled to a memory to:

(a) add a database change to a top of the database change queue in the memory; and

(b) start a transaction service thread in a non-active state, or change an existing transaction service thread in a waiting state to a non-active state, conditioned upon whether there are less than, or not less than a predetermined maximum number of transaction service threads present, respectively.

16. A medium or media in accordance with Claim 15 further having recorded thereon instructions configured to instruct the computing apparatus to remove a non-active transaction service thread conditioned upon there being more than the lesser of said predetermined maximum number or a dynamically determined optimum number of transaction service threads present, and to determine said dynamically determined optimum number of transaction service threads dependent upon a ratio of an arrival rate of database changes to the queue divided by a service time of items removed from the queue.

17. A medium or media in accordance with Claim 15 further having recorded thereon instructions configured to instruct the computing apparatus to:

change the state of a non-active transaction service thread to active conditioned upon there being a database change in the queue; and

using the active transaction service thread:

remove a bottom database change from the queue;

perform database changes specified by the removed database change; and

place the transaction service thread into the non-active state.

18. A medium or media in accordance with Claim 15 also having recorded thereon instructions configured to instruct the computing apparatus to further condition said adding a non-active transaction service thread upon there being less than a dynamically determined optimum number of

transaction service threads, and to determine said dynamically determined optimum number of transaction service threads dependent upon a ratio of an arrival rate of database changes to the queue divided by a service time of items removed from the queue.

19. A medium or media in accordance with Claim 15 further having recorded thereon instructions configured to instruct the computing apparatus to add a corresponding set of one or more interested listeners to the top of said queue along with said database change.

20. A medium or media in accordance with Claim 19 further having recorded thereon instructions configured to instruct the computing apparatus to:

change the state of a non-active transaction service thread to active conditioned upon there being a database change in the queue; and

using the active transaction service thread:

remove a bottom database change and the corresponding set of interested listeners from the queue;

notify said interested listeners that the removed database change has begun;

perform and committing database changes specified by the removed database change, conditioned upon obtaining locks necessary for transactions required for the removed database change;

notify said interested listeners of a completion status of the removed database change; and

place the transaction service thread into the non-active state.